



E.T.S. de Ingeniería Industrial,  
Informática y de Telecomunicación

# DESARROLLO DE UN JUEGO DE PREGUNTAS BASADO EN UN BOT ENFOCADO A LA EDUCACIÓN EN SALUD

---



Grado en Ingeniería en Tecnologías de  
Telecomunicación

## Trabajo Fin de Grado

Álvaro Cubillo Pascual

Jesús Daniel Trigo Vilaseca

Victoria Duro Suárez

Pamplona, 28 de Junio de 2017



# DESARROLLO DE UN JUEGO DE PREGUNTAS BASADO EN UN BOT ENFOCADO A LA EDUCACIÓN EN SALUD

AUTOR: Álvaro Cubillo Pascual  
DIRECTORES: Jesús Daniel Trigo Vilaseca  
Victoria Duro Suárez

Dpto. Ingeniería Eléctrica y Electrónica  
Servicio de Medicina Interna del Complejo Hospitalario de  
Navarra

UPNA  
Junio de 2017



## RESUMEN

---

Este Trabajo Fin de grado trata sobre la implementación de un bot para Telegram enfocado a la educación en el área de la salud, centrándose en las ETS (Enfermedades de transmisión sexual), destacando especialmente el VIH (Virus de la inmunodeficiencia humana), ya que hoy en día éste supone una pandemia según la OMS (Organización Mundial de la Salud).

Esta aplicación consiste en un juego de preguntas que se dividen en sucesivos niveles. Los usuarios deberán contestar correctamente todas las cuestiones de cada nivel para poder avanzar hasta el siguiente. Una vez superadas con éxito se habrá finalizado el juego. Destacar que se pueden seguir los progresos mediante unas estadísticas proporcionadas por la aplicación.

Se pretende que a través de esta herramienta el paciente esté mejor informado sobre su enfermedad. Como el proceso de aprendizaje es mediante la superación de un juego se consigue captar el interés con mayor facilidad.

## PALABRAS CLAVE

---

Bot, juego, programa, aplicación, Telegram, Heroku, DropBox, PHP, PostgreSQL, preguntas, salud, enfermedades de transmisión sexual, ETS, VIH, SIDA.

## ABSTRACT

---

This end of degree project deals with the implementation of a bot for Telegram focused on education in the area of health, concentrating on STD (sexually transmitted diseases), with particular emphasis on HIV (human immunodeficiency virus), because nowadays it supposes a pandemic according to WHO (World Health Organization).

This application is a game of questions formed by several levels. Users must answer correctly all questions at each level in order to pass to the next. Once all questions have been answered correctly, the game will be finished. It highlights that progress can be monitored through statistics provided by the application.

It is intended that with this tool the patient is better informed about his illness. As the process of learning is through the overcoming of a game, the user's interest is captured more easily.

## KEY WORDS

---

Bot, game, application, Telegram, Heroku, DropBox, PHP, PostgreSQL, questions, health, sexually transmitted diseases, STD, HIV, AIDS.



<b>1. INTRODUCCIÓN .....</b>	<b>13</b>
<u>1.1. MÉDICA.....</u>	13
<u>1.1.1. ENFERMEDADES DE TRANSMISIÓN SEXUAL.....</u>	13
<u>1.1.2. EDUCACIÓN EN SALUD .....</u>	16
<u>1.1.3. GAMIFICACIÓN .....</u>	16
<u>1.2. TECNOLÓGICA .....</u>	18
<u>1.2.1 BOTS .....</u>	18
<u>1.2.2 BOT DE TELEGRAM .....</u>	19
<u>1.3. MOTIVACIÓN Y OBJETIVOS .....</u>	24
<u>1.4. ESTRUCTURA DE LA MEMORIA .....</u>	25
<b>2. DEFINICIÓN DEL JUEGO.....</b>	<b>27</b>
<b>3. ARQUITECTURA PROPUESTA .....</b>	<b>29</b>
<u>3.1. FLUJO DE INFORMACIÓN .....</u>	29
<u>3.1.1. FUNCIONAMIENTO.....</u>	29
<u>3.1.2. COMANDOS.....</u>	30
<u>3.1.3. TECLADO.....</u>	31
<u>3.2. ACTORES.....</u>	32
<u>3.2.1. USUARIO.....</u>	32
<u>3.2.2. TELEGRAM.....</u>	33
<u>3.2.3. SERVIDOR .....</u>	35
<u>3.3. CONFIGURACIÓN .....</u>	39
<u>3.3.1. HEROKU .....</u>	39
<u>3.3.2. DROPBOX.....</u>	41
<u>3.3.3. WEBHOOK .....</u>	42
<u>3.3.4. BASE DE DATOS .....</u>	42
<u>3.3.5. POSTICO.....</u>	44
<u>3.3.6. ARCHIVOS UTILIZADOS PARA PROGRAMAR EL BOT.....</u>	45
<u>3.3.7. APLICAR LOS CAMBIOS .....</u>	45
<u>3.3.8. AÑADIR POSTGRESQL AL CÓDIGO .....</u>	46
<u>3.3.9. MOSTRAR COMANDOS.....</u>	47
<b>4. RESULTADOS .....</b>	<b>49</b>
<b>5. CONCLUSIONES .....</b>	<b>55</b>
<u>5.1. MÉDICAS.....</u>	55
<u>5.2. TECNOLÓGICAS.....</u>	55
<b>6. TRABAJOS FUTUROS .....</b>	<b>57</b>
<b>7. BIBLIOGRAFÍA.....</b>	<b>59</b>



## ÍNDICE DE FIGURAS

---

Figura 1 – Bot ELIZA [8] .....	19
Figura 2 – Gmail Bot .....	20
Figura 3 – TweetItBot .....	21
Figura 4 – Feed Reader Bot .....	21
Figura 5 – BabelGram .....	22
Figura 6 – HealthTap .....	22
Figura 7 – GameBot .....	23
Figura 8 – Dragon Islands MMO RPG .....	23
Figura 9 – Disposición de teclado .....	27
Figura 10 – Escritura de comandos .....	27
Figura 11 – Arquitectura propuesta .....	29
Figura 12 – Lista de comandos .....	30
Figura 13 – Lista de comandos al introducir “/” .....	31
Figura 14 – Ejemplos de Custom Keyboards .....	31
Figura 15 – Ejemplos de Inline Keyboards .....	32
Figura 16 – Interfaz de botupna .....	33
Figura 17 – BotFather .....	34
Figura 18 – Diagrama de flujo .....	36
Figura 19 – Estructura de la base de datos .....	37
Figura 20 – Vista previa de “botupnatable” .....	37
Figura 21 – Vista previa de “botupnatable” .....	38
Figura 22 – Registro en Heroku .....	39
Figura 23 – Menú Desplegable donde se encuentra “Dashboard” .....	39
Figura 24 – Menú desplegable donde se encuentra “Create new app” .....	40
Figura 25 – Pantalla de creación de la aplicación .....	40
Figura 26 – Botón “Settings” .....	40
Figura 27 – Dominio de la aplicación .....	40
Figura 28 – Dropbox ligado a Heroku .....	41
Figura 29 – Ruta de aplicaciones de Heroku en Dropbox .....	41
Figura 30 – Selección de Heroku Postgres .....	42
Figura 31 – Botón “Install Heroku Postgres” .....	43
Figura 32 – Seleccionar aplicación .....	43
Figura 33 – Pestaña Resources .....	43
Figura 34 – Botón “Heroku Postgres :: Database” .....	44
Figura 35 – Botón “View Credentials” .....	44
Figura 36 – Información de la base de datos .....	44
Figura 37 – Configuración de la conexión .....	45
Figura 38 – Pestaña Deploy .....	46
Figura 39 – Botón Deploy .....	46
Figura 40 – Lista de comandos .....	47
Figura 41 – Seleccionar “setcommands” .....	47
Figura 42 – Elección del bot para añadir la lista de comandos .....	47
Figura 43 – Información para introducir la lista de comandos .....	48

Figura 44 – Inicio del juego.....	50
Figura 45 – Acierto de pregunta.....	50
Figura 46 – Fallo de pregunta.....	50
Figura 47 – Fin de nivel acertando pregunta habiendo fallado anteriormente .....	50
Figura 48 – Fallo de última pregunta de nivel .....	51
Figura 49 – Acierto de todas las preguntas de un nivel .....	51
Figura 50 – Juego completado .....	51
Figura 51 – Envío de "/A", "/B", "/C" o "/D" una vez finalizado el juego .....	51
Figura 52 – Envío de "/ayuda" en cualquier momento del juego .....	52
Figura 53 – Envío de "/ayuda" una vez finalizado el juego.....	52
Figura 54 – Envío de "/estadísticas" en cualquier momento del juego.....	52
Figura 55 – Envío de "/estadísticas" una vez finalizado el juego.....	52
Figura 56 – Envío de un comando no válido .....	53





## 1. INTRODUCCIÓN

---

### 1.1. MÉDICA

#### 1.1.1. ENFERMEDADES DE TRANSMISIÓN SEXUAL

Las ETS son enfermedades que se transmiten de una persona a otra durante las relaciones sexuales. Son bastante comunes y en ocasiones la persona afectada puede no presentar síntomas.

Las más comunes son [1-4]:

- CLAMIDIA

La infección por clamidia es una ETS que puede afectar tanto a los hombres como a las mujeres. La mayoría de los infectados no presentan síntomas. La clínica más frecuente es secreción genital, disuria o dolor testicular. Puede causar daños al aparato reproductor femenino y hacer más difícil el embarazo o que este sea ectópico (fuera del útero). Se puede contraer al tener relaciones sexuales anales, vaginales u orales con una persona que tenga la infección.

- GONORREA

Es una infección que afecta tanto a los hombres como a las mujeres. La mayoría de mujeres no presentan clínica y algunos hombres tampoco. Puede localizarse en genitales, recto o garganta. Los síntomas más frecuentes son disuria, secreción genital, sangrado vaginal, prurito o dolor anal. Esta infección es más frecuente personas de 15 a 24 años.

- HEPATITIS

Infección vírica que afecta al hígado. Existen varios tipos, 3 de los cuales pueden contraerse mediante relaciones sexuales. Las hepatitis B y C se contraen por contacto con fluidos corporales, aunque en el caso de la hepatitis C el contagio vía sexual es menos frecuente. La hepatitis A se contrae por vía fecal-oral, por lo que se puede transmitir en determinadas prácticas sexuales. Según el tipo de hepatitis la clínica puede ser diferente y esta se presenta próxima a la infección o permanecer latente durante largos periodos de tiempo.

- [HERPES](#)

Es una infección causada por dos tipos de virus (herpes simple tipo 1 y 2). Con frecuencia no aparecen síntomas. Suele manifestarse como vesículas agrupadas en genitales, ano o boca que pueden ser dolorosas y es posible que se reproduzcan cada cierto tiempo (brotes). Se transmite al tener relaciones sexuales, orales, vaginales o anales con una persona infectada.

- [VIRUS DEL PAPILOMA HUMANO](#)

Es la ETS más común, ya que la mayoría de las personas sexualmente activas lo contraen en algún momento de su vida. Existe muchos serotipos, algunos de ellos pueden predisponer a padecer cáncer. La mayoría de las personas no presentan síntomas y en algún caso causan verrugas genitales. No se suele buscar la infección por papiloma sino sus consecuencias, como lesiones precancerosas en el cuello uterino. Se contrae mediante contacto oral, vaginal o anal con una persona portadora. Existe una vacuna que previene algunos de los serotipos.

- [VAGINOSIS BACTERIANA](#)

Existencia de un exceso de distintas bacterias en la vagina. La clínica más frecuente es aumento de secreción vaginal con fuerte olor a pescado, disuria o prurito local. No es clara la forma de transmisión pero se da en mujeres sexualmente activas y facilita la infección por otras ETS.

- [PIOJOS PÚBICOS](#)

También llamados “ladillas”. Son pequeños parásitos que se adhieren a la piel y al vello del área genital. Se eliminan fácilmente con un tratamiento que se puede adquirir en la farmacia. Se produce una infección que causa intenso prurito en la zona genital a causa de la mordedura del insecto. Se contrae por contacto genital con una persona portadora.

- [SARNA O ESCABIOSIS](#)

Infección que no es únicamente de transmisión sexual, aunque ésta sí es una vía frecuente. Es causada por un ácaro que se suele localizar en muñecas, dedos o nalgas y que produce una erupción cutánea pruriginosa.

- SÍFILIS

ETS que en su fase inicial es de fácil tratamiento, pero que si no se trata puede causar complicaciones a largo plazo, incluso la muerte. Existen 4 fases en la infección: primaria (lesión aislada no dolorosa y auto limitada en el sitio de contagio), secundaria (erupción cutánea que afecta a palmas y plantas que puede acompañarse de clínica sistémica y también es auto limitada), latente (persiste la infección pero no presenta clínica) y avanzada (solo se da en una parte de los no tratados y causa afectación nerviosa y daño en órganos internos). Se contrae mediante relaciones sexuales anales, vaginales u orales o durante el parto al recién nacido.

- TRICOMONIASIS

Es una infección causada por un protozoo. El 70% de los infectados no presenta clínica. Es una causa importante de vaginitis en mujeres y de uretritis en hombres. Se transmite por contacto durante la relación sexual.

- VIH/SIDA

El VIH (Virus de la Inmunodeficiencia Humana) es un microorganismo que ataca al sistema inmune de las personas, debilitándolo. La transmisión del virus se produce por el contacto con diversos fluidos corporales. El Síndrome de la Inmunodeficiencia Adquirida o SIDA es el estadio más avanzado de la infección por el VIH, caracterizado por la aparición de una serie de infecciones (oportunistas) que sólo se dan en personas que tienen un sistema inmune deficiente.

EL VIH es un importante problema de salud pública mundial. Según la OMS (Organización Mundial de la Salud), la prevalencia mundial de la infección en 2015 era de 36,7 millones de personas y en España era de casi 35 mil. Hasta este momento el VIH se ha cobrado más de 35 millones de vidas, de las cuales 1.1 millones fueron en 2015 [5].

No existe aún cura para la infección por VIH pero existen fármacos (antirretrovirales) que pueden controlar el virus y ayudar a prevenir su transmisión. La adhesión al tratamiento de los pacientes afectados, el conocimiento sobre la propia enfermedad, los mecanismos de transmisión y prevención de la misma son claves a la hora de frenar la epidemia.

En 2016, la OMS aprobó una nueva estrategia contra el VIH, entre las medidas que se deberían implementar están la divulgación de información y el uso de la innovación para agilizarla [5].

### 1.1.2. EDUCACIÓN EN SALUD

Se entiende como educación en salud a las oportunidades de aprendizaje creadas conscientemente que suponen una forma de comunicación destinada a mejorar la alfabetización sanitaria, incluida la mejora del conocimiento de la población en relación con la salud y el desarrollo de habilidades personales que conduzcan a la salud individual y a la comunicación. Por lo cual, la educación en salud es una herramienta básica, tanto para la prevención de determinadas enfermedades como para su curación y/o rehabilitación, y como un instrumento indiscutible de promoción de salud

En muchas ocasiones, los pacientes no conocen adecuadamente las enfermedades que padecen ni los efectos de la medicación que toman. Mediante la educación en salud se puede conseguir una mejor adherencia terapéutica y la modificación de estilos de vida poco saludables. En particular, se ha demostrado que mediante sistemas informáticos se puede mejorar del comportamiento de los usuarios en ámbitos tan variados como la nutrición, el tabaquismo, el abuso de sustancias, o enfermedades de transmisión sexual (ETS) [6-7].

### 1.1.3. GAMIFICACIÓN

La Gamificación es una técnica de aprendizaje que traslada la mecánica de los juegos al ámbito educativo-profesional con el fin de conseguir mejores resultados, ya sea para comprender mejor algunos conocimientos o mejorar alguna habilidad entre otros muchos objetivos.

El modelo de juego pretende motivar a los participantes, para ello se utilizan una serie de técnicas mecánicas y dinámicas.

- TÉCNICAS MECÁNICAS

Las técnicas mecánicas son las formas de recompensar al usuario en función de los objetivos alcanzados. Algunas de las técnicas mecánicas más utilizadas son las siguientes:

- Acumulación de puntos:

- Se asigna un valor cuantitativo a determinadas acciones y se van acumulando a medida que se realizan.

- Escalado de niveles:

- Se definen una serie de niveles que el usuario debe ir superando para llegar al siguiente.



-Obtención de premios:

A medida que se van consiguiendo objetivos se entregan una serie de premios.

-Regalos:

Bienes que se dan al jugador de forma gratuita al conseguir algún objetivo.

-Clasificaciones:

Clasificar a los usuarios en función de puntos u objetivos logrados.

-Desafíos:

Competición entre usuarios, el mejor obtiene una recompensa.

-Misiones o retos:

Conseguir superar un reto u objetivo.

- TÉCNICAS DINÁMICAS

Las técnicas dinámicas persiguen motivar al usuario para jugar y seguir adelante y así conseguir sus objetivos. Algunas de las más utilizadas son:

-Recompensa:

Obtener un beneficio merecido.

-Estatus:

Establecerse en un nivel jerárquico social valorado.

-Logro:

Como satisfacción o superación personal.

-Competición:

Por competir e intentar superar a los demás.

Estas técnicas descritas pueden aplicarse en el ámbito de la salud, mediante videojuegos y dinámicas de grupo, permitiendo aumentar el compromiso y la participación de los pacientes en programas de tratamiento y/o rehabilitación, así como mejorar las capacidades y aptitudes de los profesionales sanitarios.

Un entorno de aplicación interesante son las ETS, más concretamente el VIH, ya que son enfermedades complejas de gestionar debido a la gran cantidad de fármacos que se deben tomar, además de la responsabilidad social que conlleva la propia enfermedad, entre otros factores.

## 1.2. TECNOLÓGICA

### 1.2.1 BOTS

Los bots son programas informáticos que tratan de simular el comportamiento humano. A día de hoy, diversas plataformas (tales como Telegram, Facebook, Twitter, Skype, etc.) ofrecen interfaces abiertas (llamadas "*Application Programming Interfaces*", APIs) para que terceros desarrollen sus propios bots de forma que se integren en la plataforma y los usuarios puedan acceder a estos servicios. Telegram, por ejemplo, ofrece la posibilidad de crear bots que formen parte de una conversación personal o grupal y que generen interacción con los usuarios.

La historia de los bots comenzó varios años antes de que se programara el primero de ellos. Fue en los años 50 del siglo XX, cuando Alan Turing comenzó a considerar la posibilidad de que las máquinas pensasen. En 1950 diseñó el Test de Turing, un postulado teórico con el que aspiraba a poder comprobar si una máquina tiene comportamientos inteligentes similares o indistinguibles a los de un humano.

Su trabajo inspiró a varios informáticos, como Joseph Weizenbaum, que en 1966 desarrolló un programa con el que pretendía engañar a humanos haciéndoles creer que estaban hablando con una persona. Esto dio lugar al primer bot conversacional, se llamaba ELIZA [8].

ELIZA reconocía palabras clave y preguntaba sobre ellas, creando la ilusión de que había un entendimiento real. Para llegar a ese resultado Weizenbaum identificó cinco problemas técnicos fundamentales:

- La identificación de palabras clave.
- Descubrimiento de un contexto mínimo.
- La elección de transformaciones apropiadas.
- La generación de respuestas adecuadas.
- La capacidad de reaccionar ante la ausencia de palabras críticas.

La Figura 1 ilustra el funcionamiento de ELIZA.

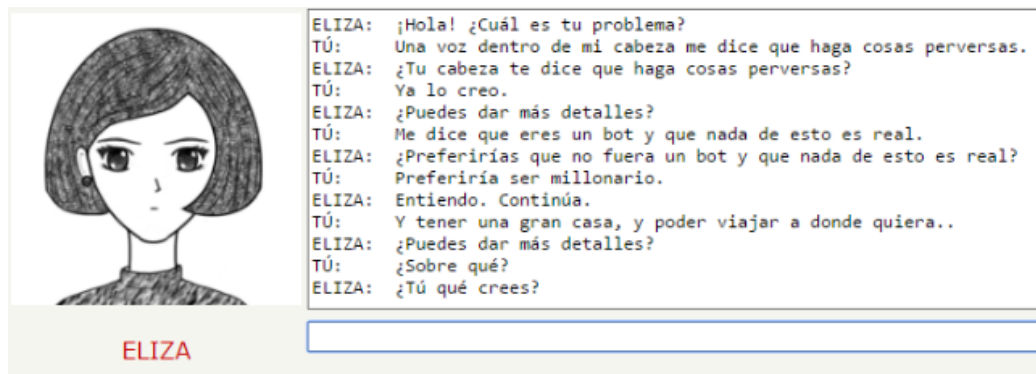


Figura 1 – Bot ELIZA [8]

Además de la faceta conversacional, los bots permiten la creación de juegos. Esto abre la puerta a la *Gamificación*.

### 1.2.2 BOT DE TELEGRAM

Aunque actualmente existen infinidad de bots para diferentes plataformas este proyecto se va a centrar en los bots de Telegram, que son aplicaciones de terceros que se ejecutan dentro de Telegram.

La comunicación con los servidores se hace a través de una sencilla interfaz HTTPS (Hypertext Transfer Protocol Secure) que ofrece una versión simplificada de la API de Telegram. Se llama a esa interfaz API de BOT [9].

Los usuarios pueden interactuar con éstos enviándoles mensajes, comandos y solicitudes en línea, se controlan mediante llamadas HTTPS a la API de BOT de Telegram, esto es lo que se conoce como WebHook, es decir, callback HTTPS de usuario.

Para que los servidores de Telegram conozcan la dirección del servidor externo donde está alojado el bot y su TOKEN (identificador utilizado para facilitar el proceso de autenticación), es necesario enviar un WebHook a la API de BOT con la URI (Uniform Resource Identifier) del servidor externo, junto con el TOKEN del bot. La petición HTTPS quedaría de la siguiente manera:

`https://api.telegram.org/bot[TOKEN]/setWebhook?url=[URL]`

Los bots de Telegram son cuentas especiales que no necesitan un número de teléfono que configurar.

Para enviar mensajes y comandos a los bots se abre un chat con ellos, también se pueden enviar solicitudes desde el campo de entrada escribiendo el nombre

de usuario del bot y la consulta a realizar, para esto es necesario que sea un tipo de bot especial llamado “inline bot”.

Los mensajes, comandos y peticiones enviadas por los usuarios se procesan en el software que se ejecuta en los servidores de Telegram. El intercambio de datos entre los servidores de Telegram y el servidor externo donde está alojado el bot se hace en formato JSON (JavaScript Object Notation), un formato de texto ligero para el intercambio de datos que describe éstos con una sintaxis dedicada que se usa para identificarlos y gestionarlos.

Los servidores manejan todo el cifrado y la comunicación con la API de Telegram de una manera transparente, por lo que no es necesario encargarse de la seguridad.

- [EJEMPLOS DE BOTS DE TELEGRAM](#)

Algunos de los bots de Telegram más interesantes que hay disponibles son:

-Gmail Bot:

Bot mediante el cual se pueden enviar y recibir emails, descargar archivos adjuntos y documentos.

La Figura 2 ilustra el funcionamiento de Gmail Bot.

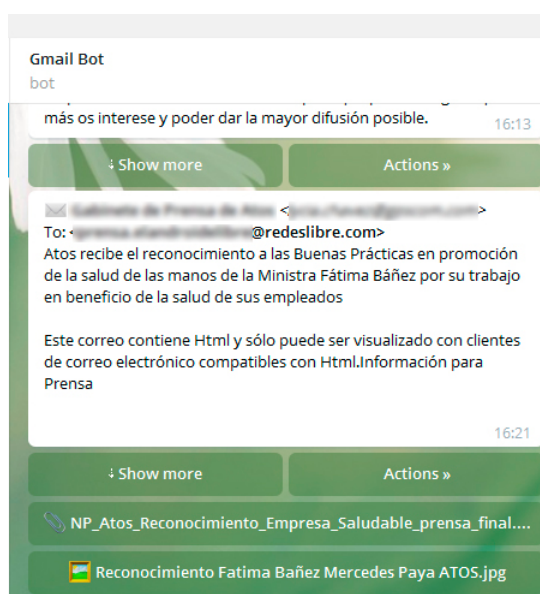


Figura 2 – Gmail Bot

#### -TweetItBot:

Es capaz de enviar y borrar tuits, retuitear, marcar favoritos y publicar imágenes.

La Figura 3 ilustra el funcionamiento de TweetItBot.

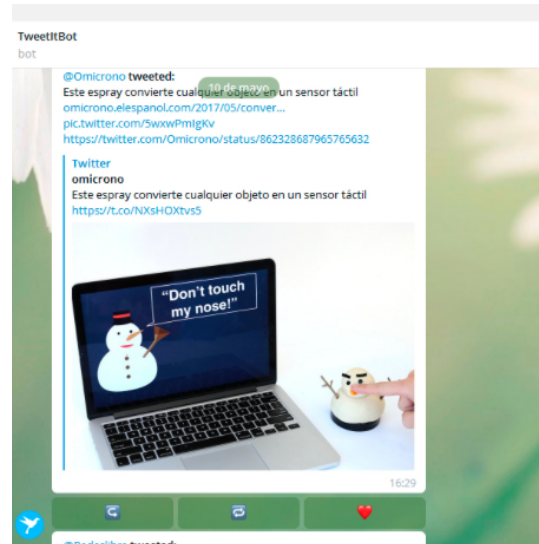


Figura 3 – TweetItBot

#### -Feed Reader Bot

Con este bot se pueden leer RSS, (Really Simple Syndication), es un formato XML (eXtensible Markup Language) para compartir contenido en la web. Se utiliza para difundir información actualizada con frecuencia a usuarios que se han suscrito a la fuente de contenido.

La Figura 4 ilustra el funcionamiento de Feed Reader Bot.

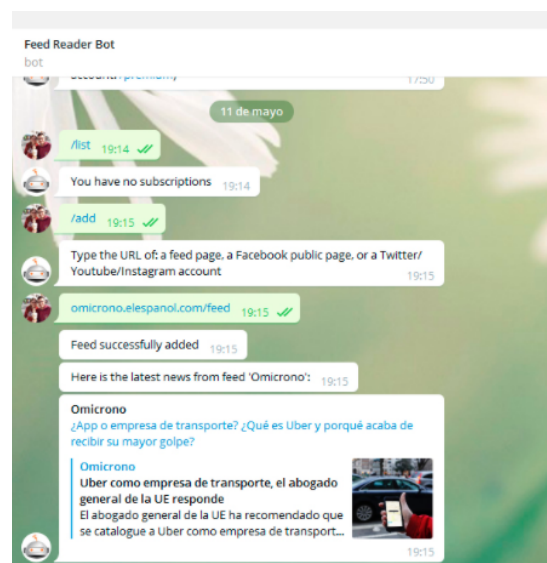


Figura 4 – Feed Reader Bot

#### -BabelGram

Sirve para traducir fragmentos de texto, funciona enviando “@BabelgramBot [idioma] [idioma] [texto]”.

La Figura 5 ilustra el funcionamiento de BabelGram.

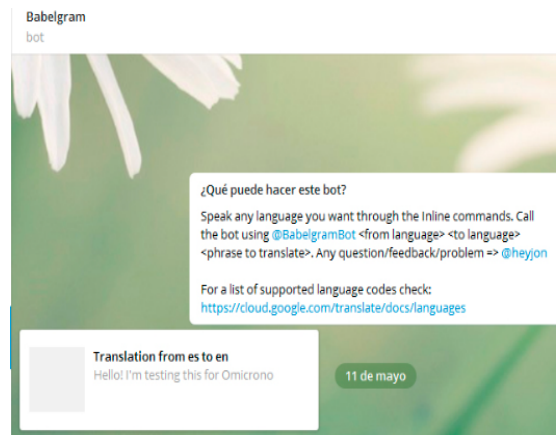


Figura 5 – BabelGram

También existen bots dedicados al área de la salud, como puede ser:

#### -HealthTap:

Proporciona consejos sobre la salud, es un bot que está disponible en varios idiomas.

La Figura 6 ilustra el funcionamiento de HealthTap.

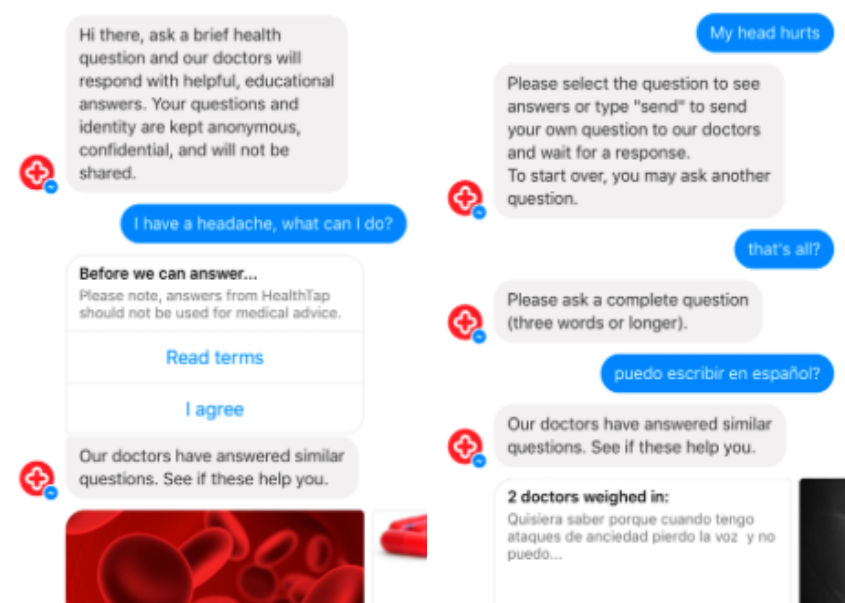


Figura 6 – HealthTap

Además, se pueden encontrar gran variedad de bots de juegos, véase:

#### -GameBot:

Es un conjunto de mini juegos desarrollados en HTML5 adaptados para jugar en dispositivos móviles. Actualmente los juegos que se encuentran disponibles son “Math Battle”, “Corsairs” y “Lumberjack”, como puede verse en la Figura 7.

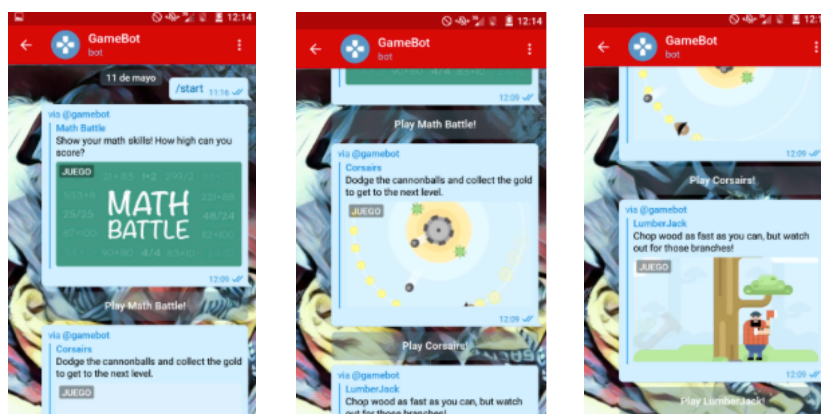


Figura 7 – GameBot

#### -Dragon Islands MMO RPG:

Es un bot en el que se implementa un videojuego de rol totalmente basado en texto. Se podrán hacer misiones por encargo de personajes, enfrentarse en batallas contra monstruos u otros jugadores, o mejorar las armas, entre otras cosas. Se puede ver una demostración de la mecánica del juego en la Figura 8.

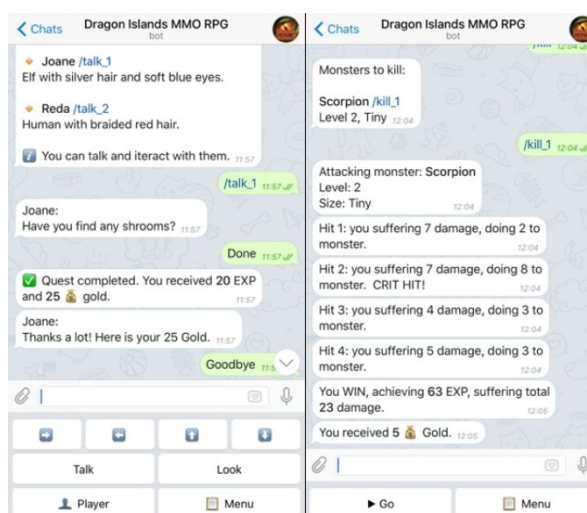


Figura 8 – Dragon Islands MMO RPG

### 1.3. MOTIVACIÓN Y OBJETIVOS

Debido a la expansión y al aumento de la velocidad de internet, al auge de las tablets y los smartphones, a la popularización de las redes sociales y programas de mensajería, y a la consolidación de los servicios ofrecidos en la nube, es viable, funcional e interesante, la creación de un bot basado en un juego que cubra la necesidad de informar a personas que padezcan alguna enfermedad. También es interesante la posibilidad de realizar un seguimiento a cada usuario en función de los resultados obtenidos.

Por estos motivos, este Trabajo Fin de Grado (TFG) tiene como objetivo desarrollar un juego de preguntas y respuestas basado en un bot de Telegram y que esté enfocado a la educación en salud. Para llevar a cabo este propósito, en el marco del TFG se diseñará la arquitectura tecnológica necesaria y se implementará la lógica del bot de forma que plantee las preguntas del juego y devuelva explicaciones razonadas de la respuesta correcta, con el objetivo de mejorar la educación en salud. Asimismo, el programa integrará unas estadísticas de progreso y ayuda al usuario.

Esto nos lleva a los siguientes objetivos particulares:

- Colaboración de la definición del juego y sus reglas, junto con los directores de proyecto.
- Diseño de la arquitectura tecnológica que permita soportar el juego.
- Implementación de dicha arquitectura.
- Diseño del modelo de datos para albergar la información.
- Implementación de dicho modelo de datos.
- Despliegue de la implementación en una plataforma en la nube.



#### 1.4. ESTRUCTURA DE LA MEMORIA

En el capítulo 2 se aborda la definición del juego, se explican sus reglas y su funcionamiento, así como las características que posee.

En el capítulo 3 se muestra la arquitectura propuesta, está subdividido en tres partes. En la primera se describe como es el flujo de la información, en la segunda los actores presentes en el proceso, y en la tercera se explican todos los pasos necesarios de configuración para el correcto funcionamiento del bot.

En el capítulo 4 se presentan capturas de pantalla de los resultados obtenidos, se pretende enseñar el funcionamiento del bot de una manera visual.

En el capítulo 5 se exponen las conclusiones tanto médicas como tecnológicas a las que se ha llegado una vez finalizado el TFG.

En el capítulo 6 se proponen posibles modificaciones del programa, se expone la posibilidad de probarlo en un entorno real y recalca su escalabilidad.

En el capítulo 7 se lista la bibliografía consultada para el desarrollo de este TFG.



## 2. DEFINICIÓN DEL JUEGO

---

Al crear una nueva conversación con el bot se empieza un juego de preguntas que están clasificadas en niveles según su dificultad.

Si el usuario juega por primera vez se le añade a una tabla de la base de datos.

Las preguntas han sido diseñadas por personal médico del servicio de medicina interna del Complejo Hospitalario de Navarra (*Victoria Duro Suárez*) y se encuentran almacenadas en otra tabla de la base de datos. De ser necesario se pueden modificar, añadir o eliminar.

Cada pregunta que envíe el bot tendrá cuatro opciones posibles y sólo habrá una correcta.

La elección de la pregunta correcta se puede hacer de dos maneras, seleccionando la opción desde un teclado que consta de cuatro botones, dispuestos como se muestra en la figura 9, o bien escribiendo directamente los comandos “/A”, “/B”, “/C” o “/D” en el campo de entrada, como puede verse en la figura 10.

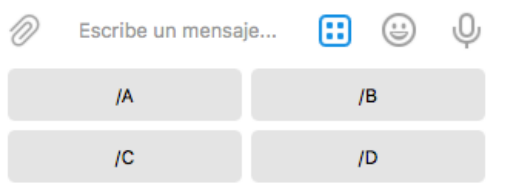


Figura 9 – Disposición de teclado



Figura 10 – Escritura de comandos

Si se aciertan todas las preguntas del nivel se pasará al siguiente, si se falla alguna se repetirá el nivel.

Cada vez que se conteste a una pregunta el bot informará si se ha acertado o no, y dará una explicación si fuera necesario.

Al llegar al final ya no se enviarán más preguntas.

Se podrán ver las estadísticas del jugador cuando se desee y en caso de tener alguna duda con el funcionamiento del bot se podrá consultar la ayuda.



### 3. ARQUITECTURA PROPUESTA

La arquitectura que se ha utilizado se muestra en la figura 11.

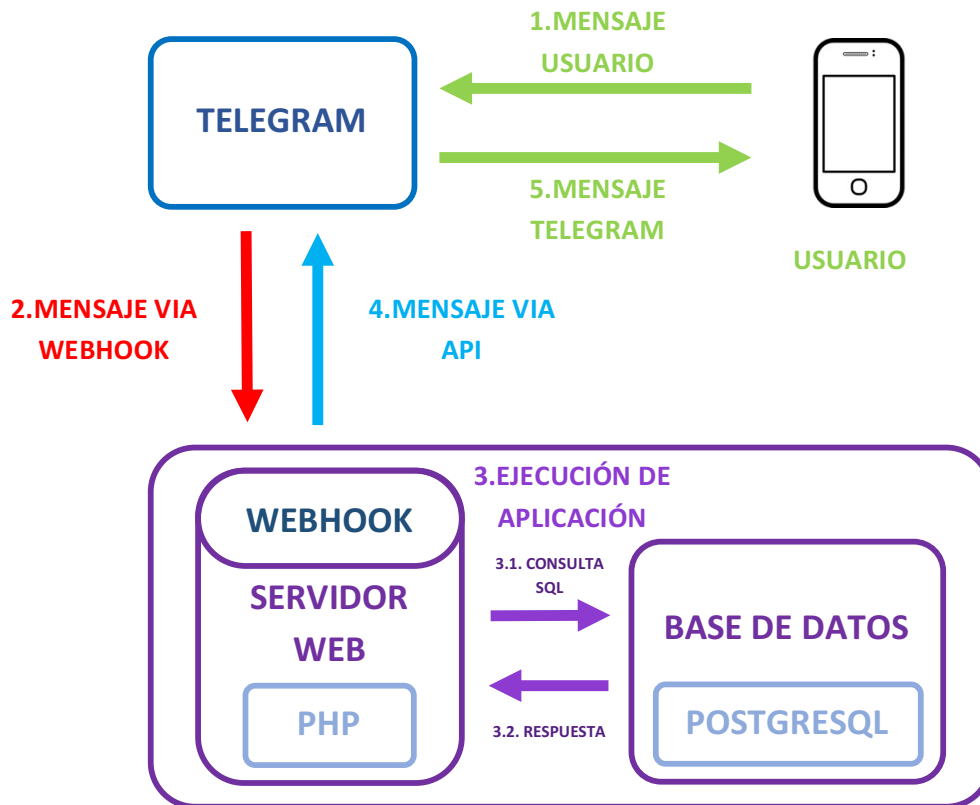


Figura 11 – Arquitectura propuesta

#### 3.1. FLUJO DE INFORMACIÓN

##### 3.1.1. FUNCIONAMIENTO

Los pasos que sigue un mensaje desde que es enviado por el usuario hasta que éste recibe la contestación son los siguientes [10]:

1. El usuario envía un mensaje de texto con el comando deseado al servidor de Telegram, éste lo procesa y obtiene una salida en formato JSON.
2. Telegram envía dicho JSON al servidor donde se aloja la aplicación web mediante un webhook (se envía una solicitud HTTP a una URI).

3. La aplicación web se ejecuta en función del comando recibido, se establece una conexión con la base de datos (3.1. Se envía una consulta y 3.2. se recibe la respuesta). Una vez termina la ejecución se genera una salida.
4. Dicha salida se envía por medio de la API de BOT al servidor de Telegram (se crea un objeto y se utiliza uno de sus atributos para realizar el envío).
5. Telegram envía un mensaje con la respuesta al usuario.

### 3.1.2. COMANDOS

Se han implementado tres comandos que hay que introducir en el campo de entrada.

En los comandos “/ayuda” y “/estadísticas”, si se ha finalizado el juego sólo se mostrará el resultado del comando enviado, si por el contrario aún quedan preguntas sin responder se mostrará el resultado del comando junto con la pregunta actual.

Los comandos son:

1./start: Empieza el juego, se envía la primera pregunta y borra todos los progresos.

2./ayuda: Muestra la lista de comandos existentes, como se puede ver en la figura 12.

Lista de comandos:

```
/start -> Empieza el juego
/ayuda -> Muestra la lista de
comandos compatibles
/estadisticas -> Muestra el
numero de preguntas
acertadas
```

*Figura 12 – Lista de comandos*

3./estadisticas: Muestra las siguientes propiedades:

- El identificador del chat.
- El nivel.
- Las preguntas acertadas en el nivel.
- Las preguntas realizadas en el nivel.

- Preguntas totales en el nivel.
- Preguntas totales acertadas.
- Preguntas totales realizadas.
- Preguntas totales.

Para que no sea necesario escribir los comandos está configurado que al introducir "/" aparezca en pantalla la lista de éstos y sólo con hacer click se envíen automáticamente, como se puede ver en la figura 13.

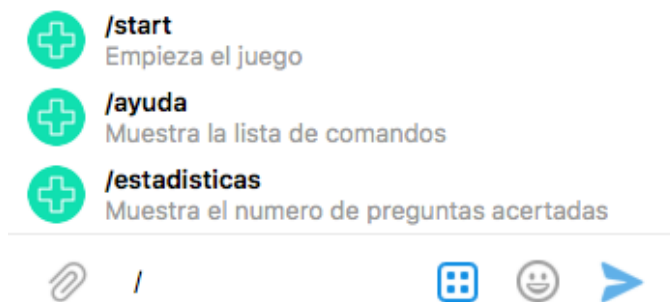


Figura 13 – Lista de comandos al introducir "/"

### 3.1.3. TECLADO

Además de los comandos, otra forma de enviar información es mediante el uso de teclados. Según la API de Telegram existen dos tipos de teclados.

1. Custom keyboards: Sirven para que el usuario utilice frases predefinidas. Al pulsar cualquiera de los botones se enviará inmediatamente el comando respectivo, con lo que se simplifica drásticamente la interacción con el bot [11].

Se pueden ver tres ejemplos de este tipo de teclado en la figura 14.

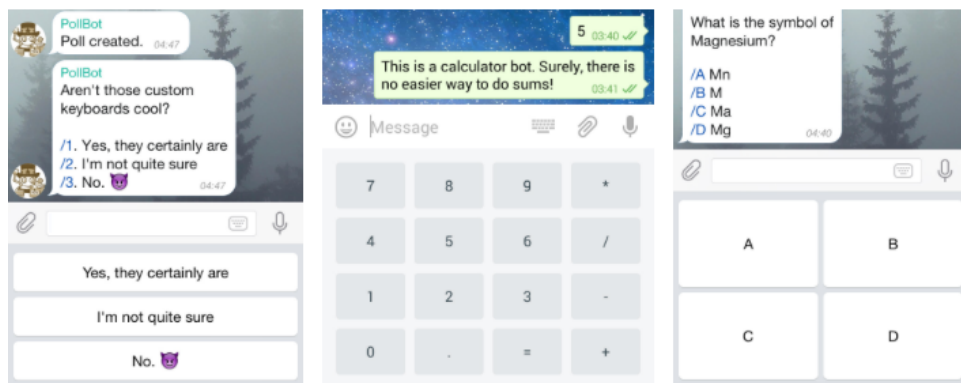


Figura 14 – Ejemplos de Custom Keyboards

2. Inline Keyboards: Sirven para realizar acciones sin enviar mensajes al chat, permiten realizar cambios en la configuración, callback buttons o URL buttons.

Se pueden ver tres ejemplos de este tipo de teclado en la figura 15.

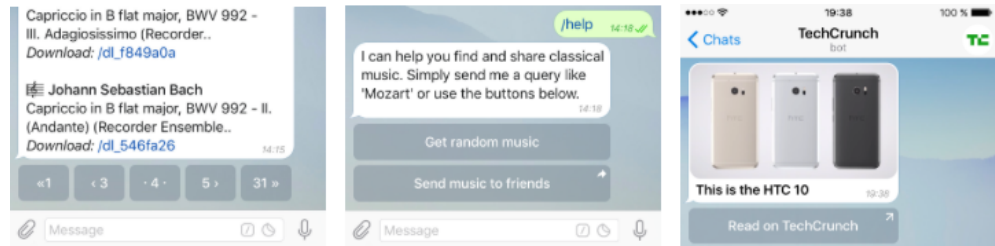


Figura 15 – Ejemplos de Inline Keyboards

En este caso se ha programado un custom keyboard, ya que se necesita enviar un comando al pulsar un botón.

## 3.2. ACTORES

A continuación, se explican los diferentes actores que se ven implicados en el proceso, siguiendo el esquema de la figura 11.

### 3.2.1. USUARIO

El usuario se instala Telegram en cualquier dispositivo compatible.

<https://telegram.org>

Se inicia una conversación con botupna, que es el nombre del bot de este proyecto, aunque puede cambiarse en cualquier momento.

[https://telegram.me/BotDePrueba\\_bot](https://telegram.me/BotDePrueba_bot)



Aparecerá la interfaz que muestra la figura 16.

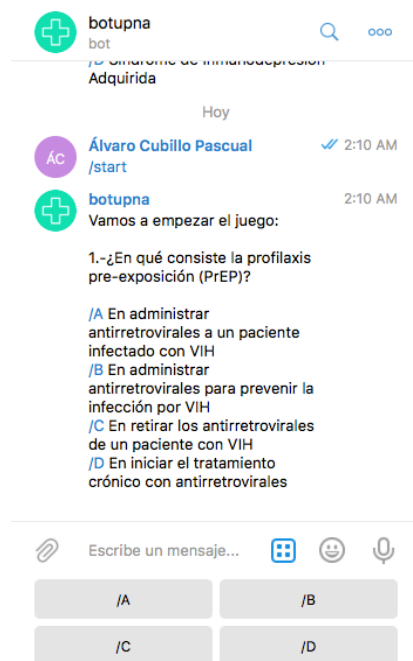


Figura 16 – Interfaz de botupna

A partir de aquí se podrá jugar siguiendo las reglas descritas en el capítulo 2 y explicadas con mayor detalle en la figura 18 del apartado 3.2.3.

### 3.2.2. TELEGRAM

Telegram es un servicio de mensajería por Internet desarrollado desde el año 2013 por los hermanos Nikolai y Pavel Durov. El servicio está enfocado en la gestión de mensajes de texto y multimedia, inicialmente fue empleado para teléfonos móviles y el año siguiente para multiplataforma.

#### 1.API de Telegram

La API de Bot está formada por una serie de URI's a través de las cuales se envían y reciben datos de una conversación de Telegram.

La comunicación entre el bot y la API se hace a través del lenguaje de marcado JSON.

El primer paso para programar un bot es obtener un token, es decir, un identificador que Telegram le asigna al bot.

Las URI's están formadas por:

- Protocolo a utilizar: *https://*
- Servidor de la API de Telegram Bot: *api.telegram.org/*
- La palabra "bot" seguida del token asignado al registro:

Bot23453242:SSASKMffdsUIOdFDNBVTYdoopfdbdydDINFhb

- El comando que se solicite.

Telegram hace de intermediario entre el usuario y el bot, encriptando la transmisión con su protocolo MTProto. Dicho protocolo encripta los mensajes simétricamente con AES-256 y la clave compartida está firmada en SHA-1.

## 2.BotFather

@BotFather, es un bot que sirve para crear y administrar bots.

Para crear un bot se inicia una conversación con el:

<https://telegram.me/botfather>

Aparecerá una interfaz como la que se muestra en la figura 17.

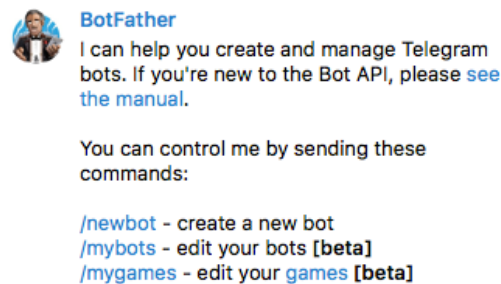


Figura 17 – BotFather

A continuación se escribe el comando **/newbot** y se asigna el nombre al bot, en este caso botupna.

Seguidamente se introduce el nombre de usuario para el bot, éste tiene que terminar obligatoriamente en bot, por ejemplo TetrisBot o tetris\_bot, en este caso BotDePrueba\_Bot. Este nombre, a diferencia del anterior no puede ser modificado.

Una vez asignados estos dos nombres BotFather proporcionará un token, el cual se guardará porque servirá para identificar al bot.

Tener en cuenta que estos pasos solo hace falta seguirlos una vez y que lo hace el desarrollador, no el usuario.

### 3.2.3. SERVIDOR

Es el núcleo de desarrollo en el TFG y está compuesto por el servidor web, donde se ejecuta el bot, y la base de datos, donde se almacena la información.

#### 1.Servidor Web

Es el programa que procesa la aplicación en el lado del servidor realizando conexiones con el cliente y generando las respuestas que este recibirá. Para transmitir los datos en este caso se utiliza el protocolo HTTPS.

Se ha programado en PHP (Hypertext Preprocessor) ya que es un lenguaje de código abierto adecuado para el desarrollo de aplicaciones web.

El diagrama de flujo mostrado en la figura 18 ilustra de una manera esquemática el funcionamiento del programa descrito en el apartado 2.

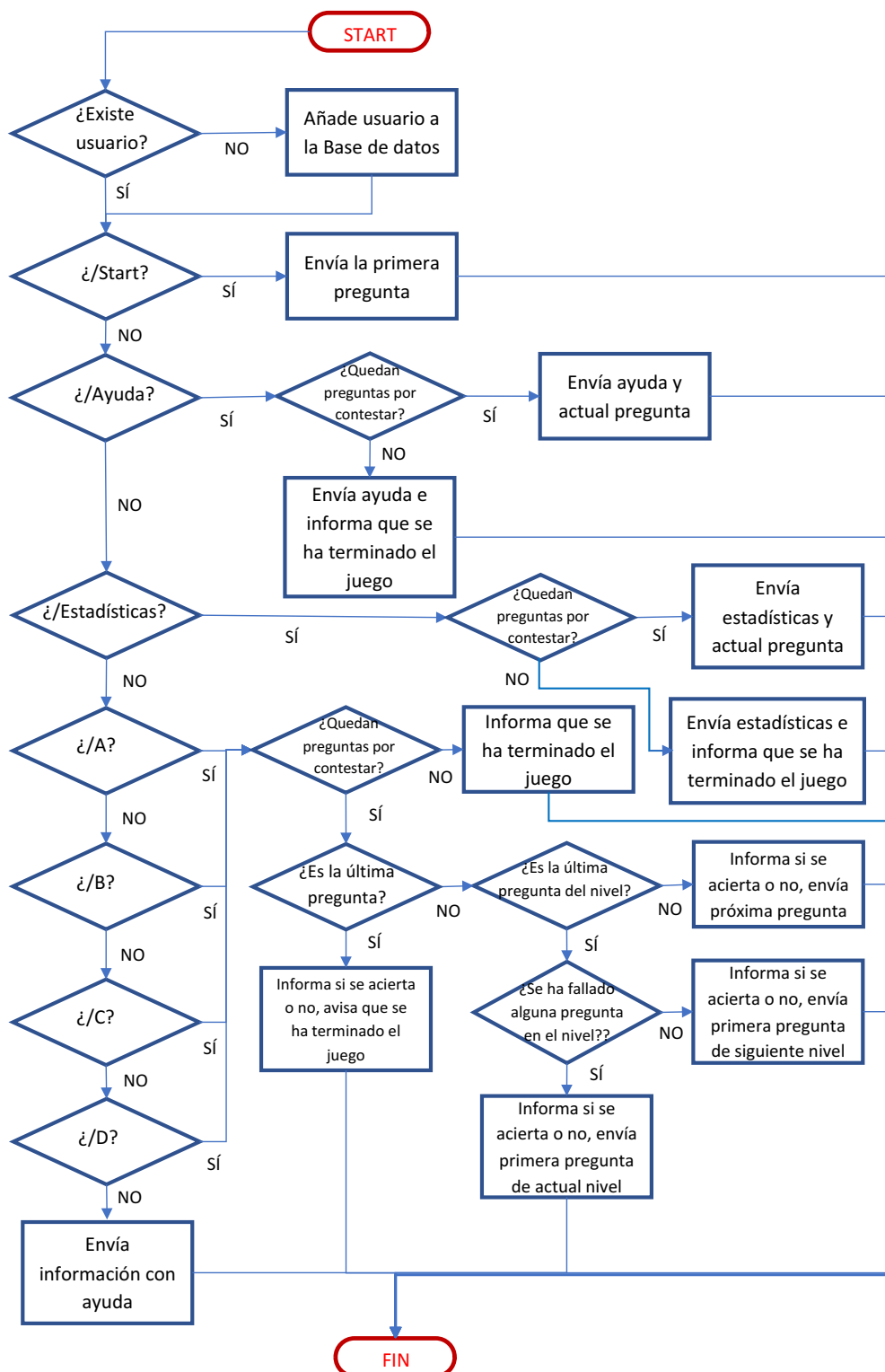


Figura 18 – Diagrama de flujo

## 2.Base de datos

Para este proyecto se ha decidido utilizar PostgreSQL, que es un sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL.

El nombre de la base de datos es “ddio92kqon2qop” y se genera automáticamente, está estructurada en dos tablas, la primera se denomina “botupnatable” y la segunda “usuariosbotupna”, como puede verse en la figura 19.

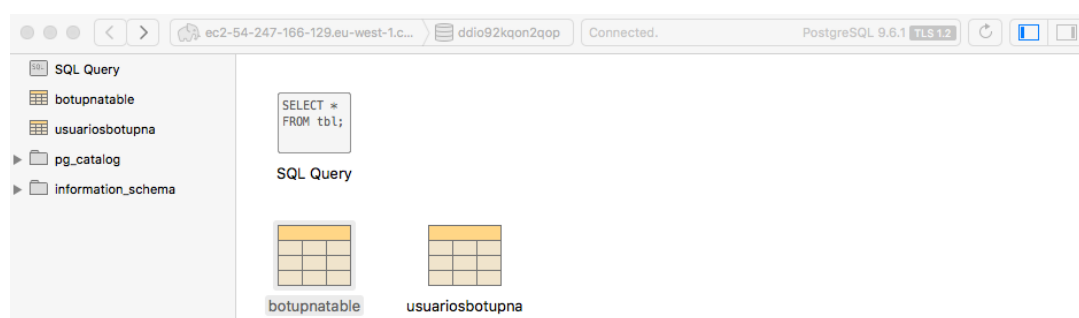


Figura 19 – Estructura de la base de datos

### 2.1. botupnatable:

La tabla consta de las siguientes columnas:

- **id**: El identificador de pregunta.
- **preguntas**: Pregunta.
- **a**: Opción A.
- **b**: Opción B.
- **c**: Opción C.
- **d**: Opción D.
- **correcta**: La opción correcta (A, B, C o D).
- **nivel**: El nivel de la pregunta en función de la dificultad.
- **explicación**: Aclaración de la respuesta.
- **categoría**: Tema sobre el que trata la pregunta.

La figura 20 muestra una imagen de la tabla “botupnatable”.

id	preguntas	a	b
1	¿En qué consiste la profilaxis pre-exposición (PrEP)?	En administrar antirretrovirales a un paciente infectado con VIH	En administrar antirretrovirales para prevenir la infección por VIH
2	La profilaxis pre-exposición (PrEP) está indicada...	como tratamiento prolongado	como método de prevención durante períodos de mayor riesgo de infección por VIH

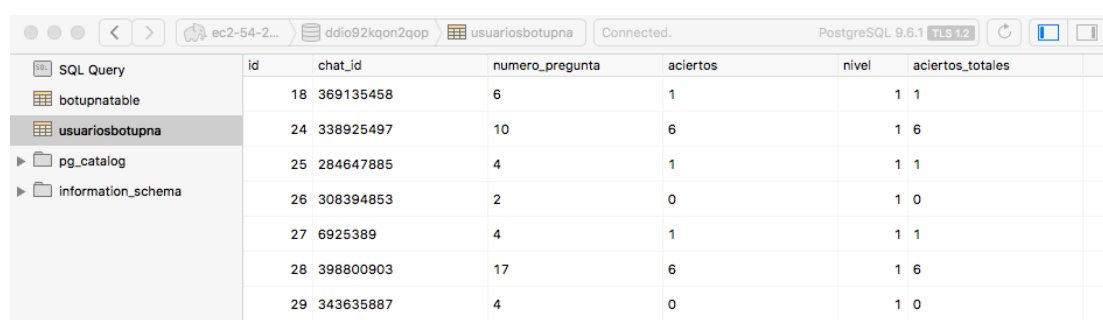
Figura 20 – Vista previa de “botupnatable”

## 2.2.usuariosbotupna:

Tabla en la que se almacenan los usuarios y sus datos:

- **chat\_id**: El identificador de usuario de la tabla.
- **chat\_id**: El identificador de usuario que proporciona Telegram.
- **numero\_pregunta**: La pregunta que debe enviar el bot.
- **aciertos**: El número de preguntas acertadas en un nivel.
- **nivel**: El nivel de las preguntas que envía el bot.
- **aciertos\_totales**: El número total de preguntas acertadas contando todos los niveles.

La figura 21 muestra una imagen de la tabla “usuariosbotupna”.



id	chat_id	numero_pregunta	aciertos	nivel	aciertos_totales
18	369135458	6	1	1	1
24	338925497	10	6	1	6
25	284647885	4	1	1	1
26	308394853	2	0	1	0
27	6925389	4	1	1	1
28	398800903	17	6	1	6
29	343635887	4	0	1	0

Figura 21 – Vista previa de “botupnatable”

## 2.3. Postico:

Para la administración de la base de datos se ha decidido utilizar “Postico”, un programa que proporciona una interfaz de usuario para gestionar las bases de datos PostgreSQL.

## 3.Heroku

Para albergar el servidor web se ha utilizado “Heroku” que es una PaaS (Platform as a Service) que soporta distintos lenguajes de programación. Fue desarrollada desde junio de 2007, con el objetivo de soportar solamente el lenguaje de programación Ruby, pero posteriormente se ha extendido el soporte a Java, Node.js, Scala, Clojure y Python y PHP.

### 3.3. CONFIGURACIÓN

En este apartado se pretende explicar el proceso de configuración de todos los programas necesarios para llevar a cabo el desarrollo del bot.

#### 3.3.1. HEROKU

Para desplegar una aplicación PHP se debe tener [12]:

- PHP instalado localmente: <http://php.net/>
- Composer instalado localmente: <https://getcomposer.org/doc/00-intro.md>

Lo primero que hay que hacer es registrarse y elegir el lenguaje de programación, en este caso PHP, como se muestra en la figura 22.

Figura 22 – Registro en Heroku

Una vez realizados estos pasos habrá que dirigirse a “Dashboard”, situado en un menú desplegable que se muestra en la figura 23.

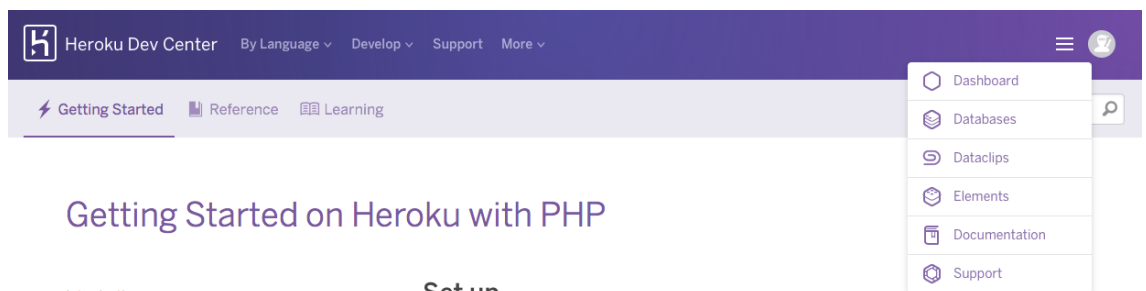


Figura 23 – Menú Desplegable donde se encuentra “Dashboard”

Al hacer click se abre una página en la que se podrá crear una nueva aplicación, en este caso el bot.

Se hace click en “New” y a continuación en “Create new app” como se muestra en la figura 24.

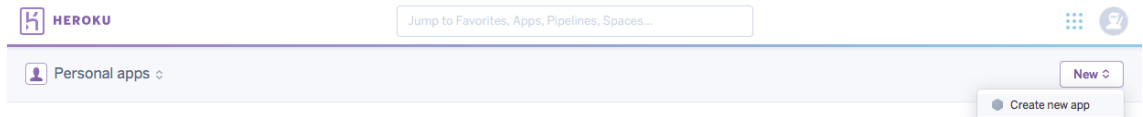


Figura 24 – Menú desplegable donde se encuentra “Create new app”

Aparecerá una pantalla en la que se introducirá el nombre de la aplicación, se elegirá el Runtime correspondiente, en este caso Europa y se hará click en “Create App” como se observa en la figura 25.

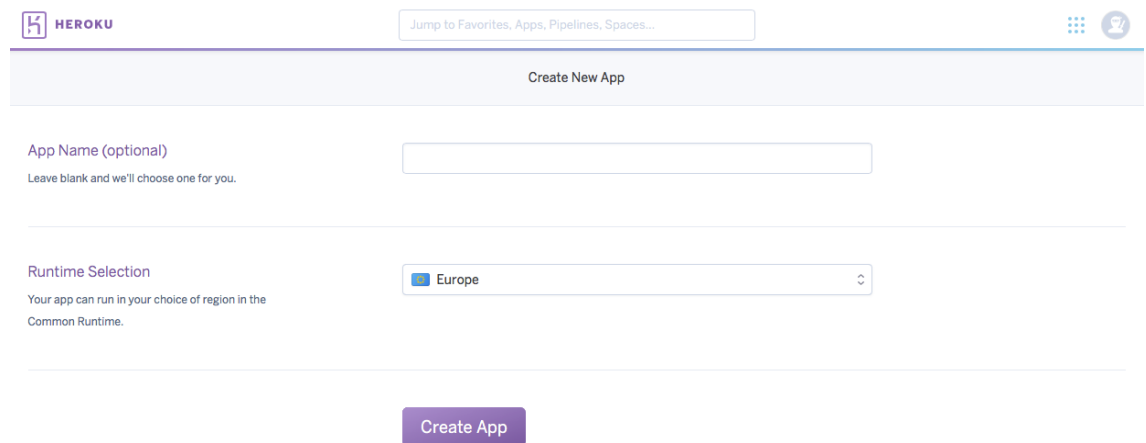


Figura 25 – Pantalla de creación de la aplicación

Ahora ya aparecerá la aplicación en “Personal apps”, accesible desde el menú “Dashboard”, se hará click en ella.

Una vez dentro se hará click en “Settings”, situado en el extremo derecho de la barra superior como muestra la figura 26.



Figura 26 – Botón “Settings”

Habrá que desplazarse hacia abajo hasta que se encuentre el dominio de la aplicación, figura 27.

## Domains and certificates

Domain Your app can be found at

Figura 27 – Dominio de la aplicación



A continuación de “Your app can be found at” se encontrará la URL, la cual se guardará porque se utilizará junto con el token.

Se ha utilizado como método de despliegue Dropbox, que es un servicio de alojamiento de archivos multiplataforma en la nube. El servicio permite almacenar y sincronizar archivos en línea y entre ordenadores y compartir archivos y carpetas con otros usuarios y con tabletas y móviles. De este modo se simplifica bastante la manera de desplegar la aplicación a Heroku.

Se hace click en “Deploy”, se elige Dropbox como método de despliegue y se ingresa el usuario y contraseña de Dropbox quedando la configuración como muestra la figura 28.

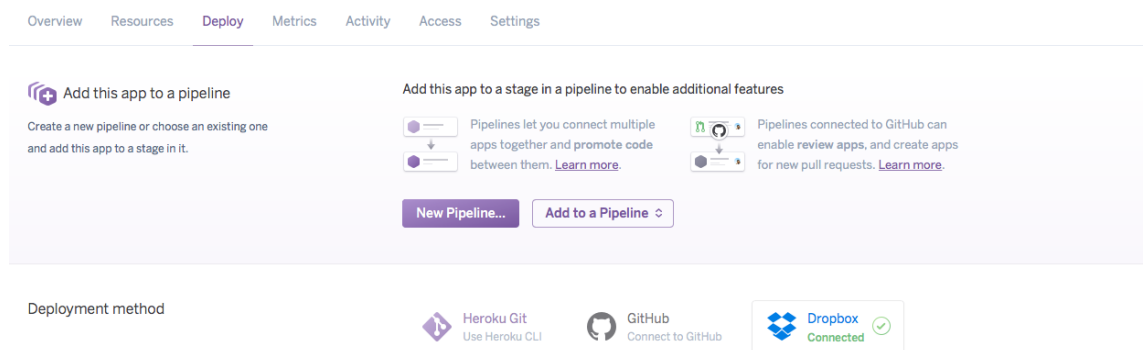


Figura 28 – Dropbox ligado a Heroku

Ya está Dropbox ligado a Heroku.

### 3.3.2. DROPBOX

Al haberse elegido en Heroku Dropbox cómo método de despliegue se habrá creado en Dropbox la ruta que se muestra en la figura 29.



Figura 29 – Ruta de aplicaciones de Heroku en Dropbox

Dentro se encontrará una carpeta con el nombre de la aplicación, es ahí donde se deberán introducir los archivos a desplegar en Heroku.

### 3.3.3.WEBHOOK

Ahora se tendrá que añadir el webhook de la app de Heroku al bot de Telegram para que tenga permisos. Para ello habrá que dirigirse al siguiente enlace [12]:

[https://api.telegram.org/bot\[TOKEN\]/setWebhook?url=\[URL\]](https://api.telegram.org/bot[TOKEN]/setWebhook?url=[URL])

Se sustituye [TOKEN] por el token que ha proporcionado botfather.

Se sustituye [URL] por la URL que se ha obtenido en Heroku.

Una vez seguidos estos pasos se verá en el navegador lo siguiente:

```
{"ok":true,"result":true,"description":"Webhook was set"}
```

### 3.3.4. BASE DE DATOS

Para poder almacenar información se necesita una base de datos. Se utilizará un add-on para Heroku.

Habrà que dirigirse a la siguiente URI:

<https://elements.heroku.com/addons/categories/data-stores>

Se elegirá Heroku Postgres, como muestra la figura 30.

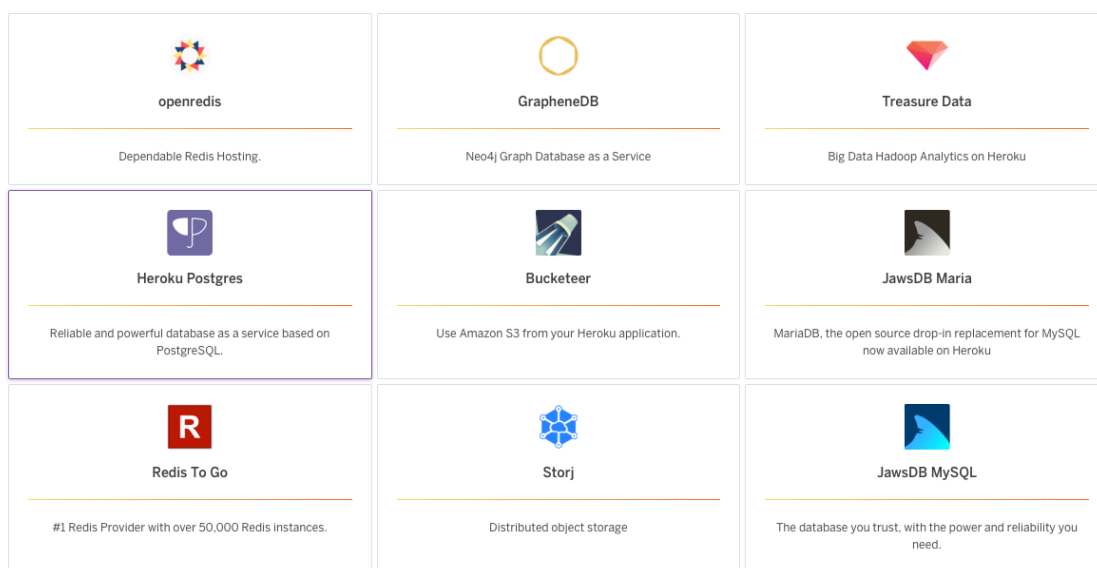


Figura 30 – Selección de Heroku Postgres

Se hará click en el botón “Install Heroku Postgres” mostrado en la figura 31.

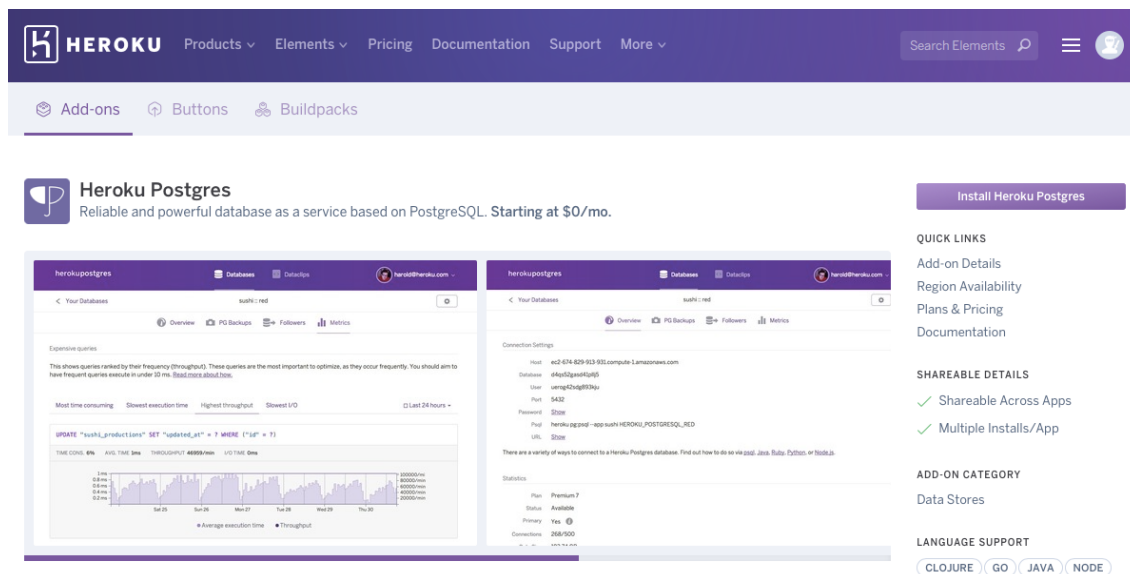


Figura 31 – Botón “Install Heroku Postgres”

Se selecciona la aplicación en la que se desee instalar el add-on, en este caso el bot, y se hace click en “Continuar”, figura 32.

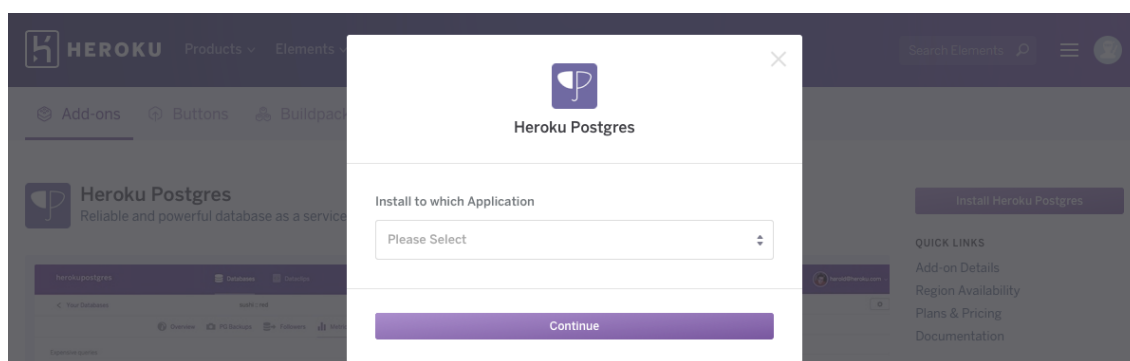


Figura 32 – Seleccionar aplicación

Ahora habrá que dirigirse a la pestaña “Resources” de la aplicación y se comprueba que se ha añadido la base de datos, figura 33.



Figura 33 – Pestaña Resources

Se hace click en el add-on que se acaba de añadir, mostrado en la figura 34.



Figura 34 – Botón “Heroku Postgres :: Database”

A continuación se hace click en el botón “View Credentials...”, figura 35.



Figura 35 – Botón “View Credentials”

Ahora se tiene visible toda la información que se necesitará posteriormente para conectar con la base de datos, figura 36.

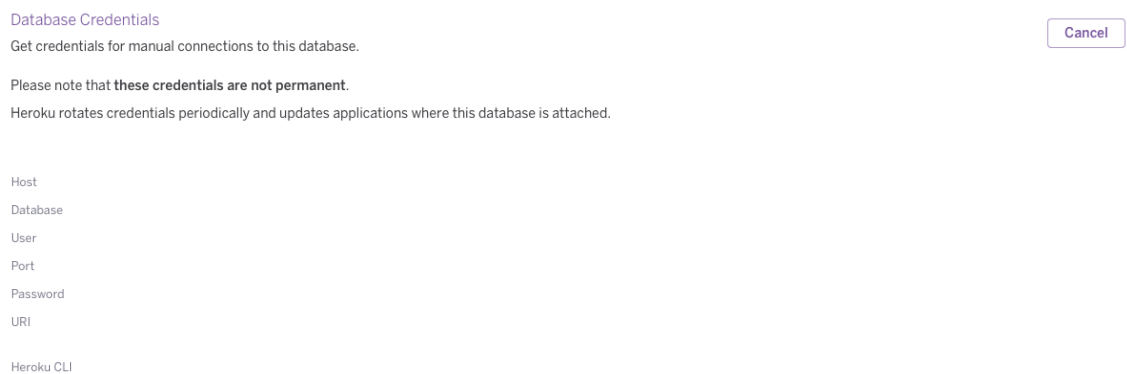


Figura 36 – Información de la base de datos

### 3.3.5. POSTICO

Para poder crear bases de datos en el proyecto habrá que conectarse al add-on “Heroku Postgres” que está instalado en la aplicación que se está diseñando, y que está alojada en los servidores de Heroku.

Se inicia Postico y se introduce la información que se ha obtenido anteriormente y que se muestra en la figura 37.

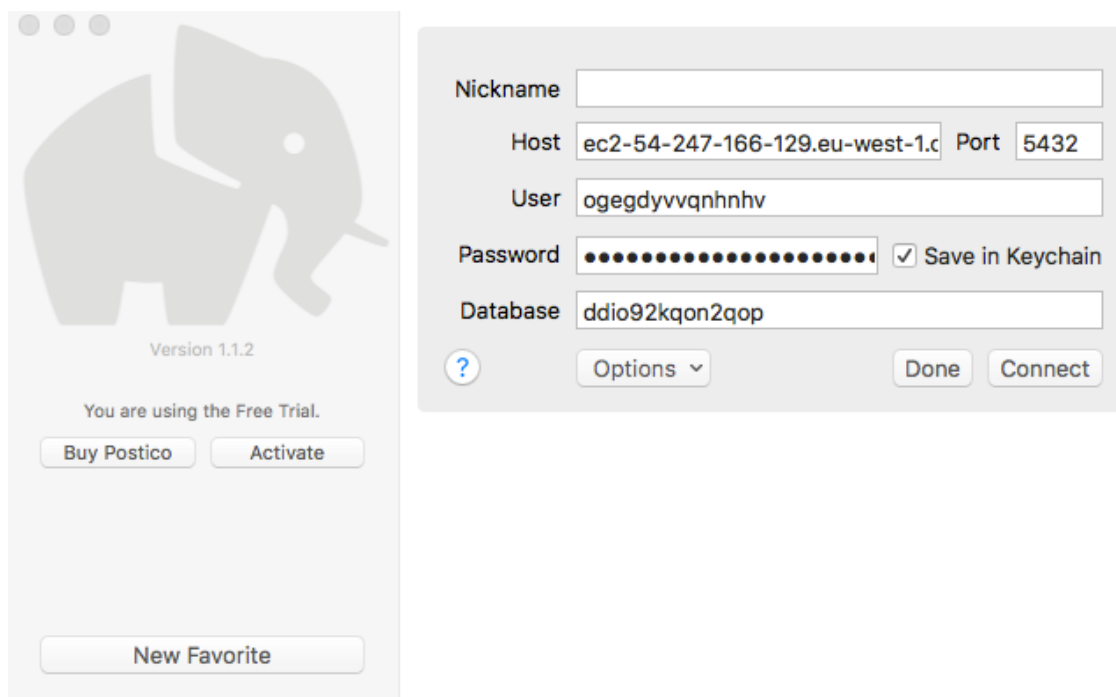


Figura 37 – Configuración de la conexión

Se hace click en “Connect” y ya se pueden crear bases de datos.

Ahora sólo es necesario rellenar los campos de las tablas de la base de datos, o bien realizar un volcado de los datos desde un archivo externo.

Para importar los datos de a tabla basta con copiar y pegar.

### 3.3.6. ARCHIVOS UTILIZADOS PARA PROGRAMAR EL BOT

Para empezar a programar el bot en PHP, se ha descargado este repositorio de git:

<https://github.com/kasramp/geekswebbot>

Proporciona los archivos necesarios para poder ejecutar un bot en Heroku.

Se copian los archivos en la ruta que se ha generado en Dropbox.

### 3.3.7. APLICAR LOS CAMBIOS

Para subir el código que se encuentra en Dropbox a Heroku se tendrá que hacer lo siguiente:

Se hace click en la aplicación en Heroku, a continuación se selecciona la pestaña “Deploy” mostrada en la figura 38.

Figura 38 – Pestaña Deploy

Una vez dentro habrá que desplazarse al final de la página, aparecerá un cuadro de texto en el que se podrá hacer una breve descripción de los cambios que se hayan realizado (Opcional), y un botón en el que pone Deploy. Al hacer click se actualizarán los archivos, como muestra la figura 39.

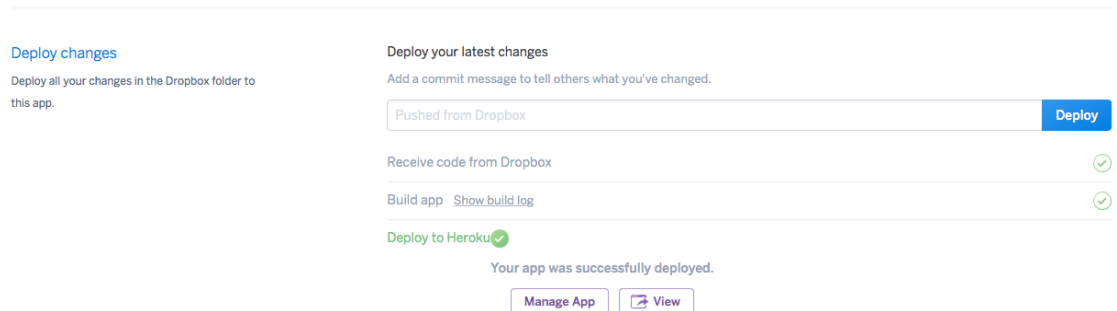


Figura 39 – Botón Deploy

Con todo esto ya se tendrá la aplicación subida a Heroku. Cada vez que se modifique código en Dropbox se tendrá que realizar este último paso.

### 3.3.8. AÑADIR POSTGRESQL AL CÓDIGO

Para poder utilizar PostgreSQL en el proyecto utilizando PHP se tendrá que editar el archivo llamado “composer.json”, para ello se abrirá el terminal del sistema operativo. Habrá que desplazarse hasta la carpeta donde está alojado el archivo y se escribirá [13]:

```
$ composer require herrera-io/silex-pdo
```

Ahora se instalará la dependencia:

```
$ composer update
```

### 3.3.9. MOSTRAR COMANDOS

Para que no sea necesario escribir los comandos se ha implementado que al introducir "/" aparezca en pantalla la lista de éstos. Sólo con hacer click en uno de ellos se envía automáticamente, figura 40.

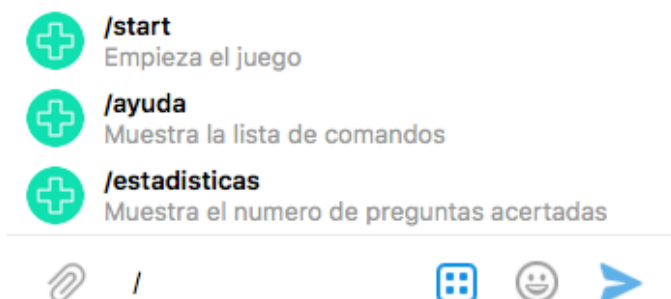


Figura 40 – Lista de comandos

La configuración se hace de la siguiente manera:

- Se abre el chat de BotFather.
- Se escribe “/setcommands”, figura 41.

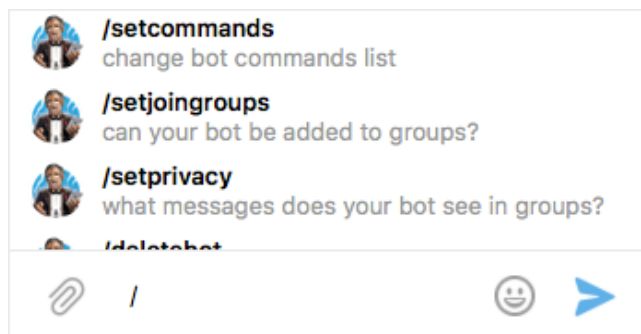


Figura 41 – Seleccionar “setcommands”

- Se elige el bot que se quiera modificar haciendo click en el, figura 42.

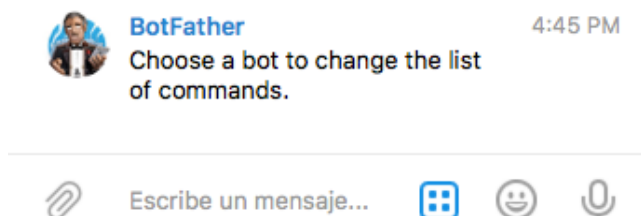
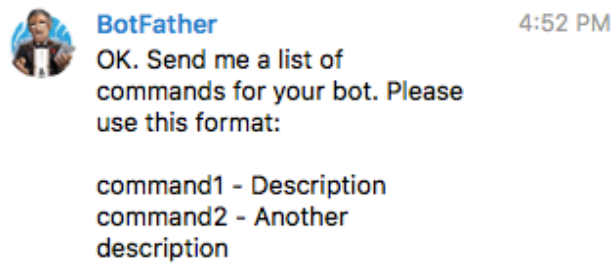


Figura 42 – Elección del bot para añadir la lista de comandos

- Ahora se tiene que introducir la lista de comandos junto a una descripción, figura 43.



*Figura 43 – Información para introducir la lista de comandos*



## 4. RESULTADOS

---

A continuación se muestran algunas capturas de pantalla del bot para que pueda verse su funcionamiento.

- En la figura 44 se muestra una captura de pantalla de lo que ocurre al enviar el comando “/start”.
- En la figura 45 se muestra una captura de lo que ocurre al acertar una pregunta.
- En la figura 46 se muestra una captura de pantalla de lo que ocurre al fallar una pregunta.
- En la figura 47 se muestra una captura de pantalla de lo que ocurre al contestar la última pregunta de un nivel correctamente, pero haber fallado alguna pregunta del mismo.
- En la figura 48 se muestra una captura de pantalla de lo que ocurre al contestar la última pregunta de un nivel erróneamente.
- En la figura 49 se muestra una captura de pantalla de lo que ocurre al contestar todas las preguntas de un nivel correctamente.
- En la figura 50 se muestra una captura de pantalla de lo que ocurre al completar el juego correctamente.
- En la figura 51 se muestra una captura de pantalla de lo que ocurre al enviar “/A”, “/B”, “/C” o “/D” una vez completado el juego.
- En la figura 52 se muestra una captura de pantalla de lo que ocurre al enviar el comando “/ayuda” en cualquier momento del juego.
- En la figura 53 se muestra una captura de pantalla de lo que ocurre al enviar el comando “/ayuda” habiendo finalizado el juego.
- En la figura 54 se muestra una captura de pantalla de lo que ocurre al enviar el comando “/estadísticas” en cualquier momento del juego.
- En la figura 55 se muestra una captura de pantalla de lo que ocurre al enviar el comando “/estadísticas” al haber finalizado el juego.
- En la figura 56 se muestra una captura de pantalla de lo que ocurre al enviar cualquier texto no reconocido por el bot.

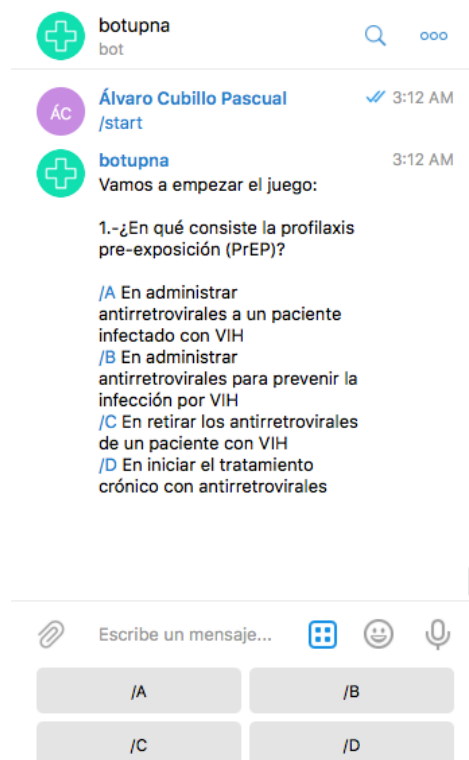


Figura 44 – Inicio del juego

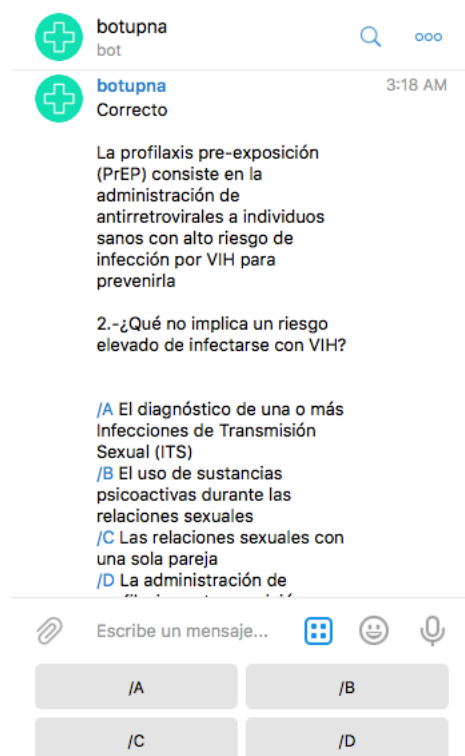


Figura 45 – Acierto de pregunta

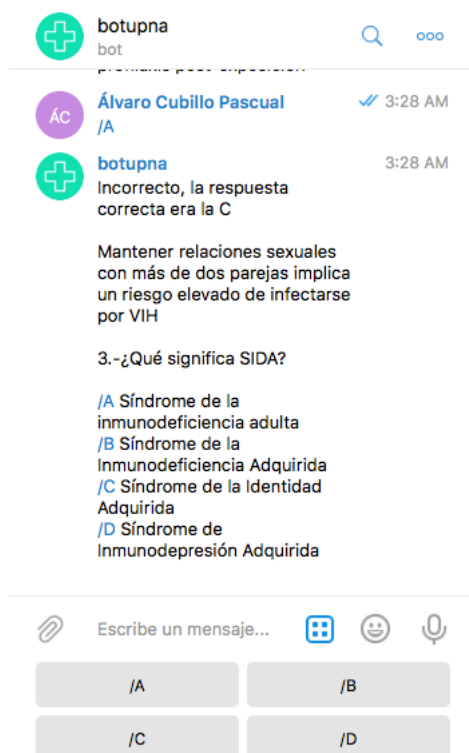


Figura 46 – Fallo de pregunta

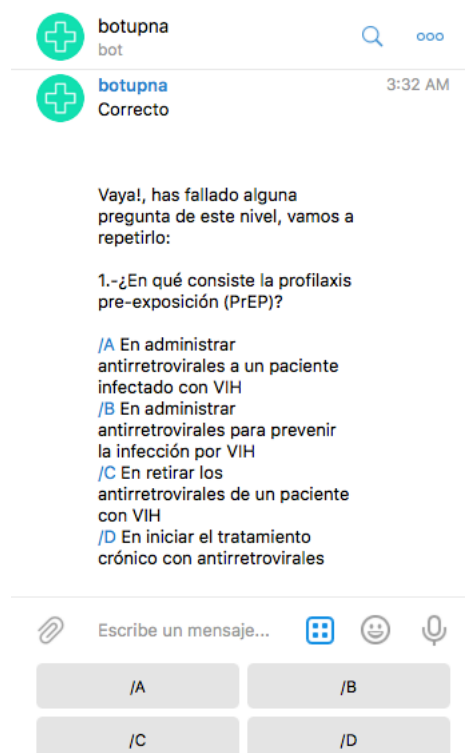


Figura 47 – Fin de nivel acertando pregunta habiendo fallado anteriormente

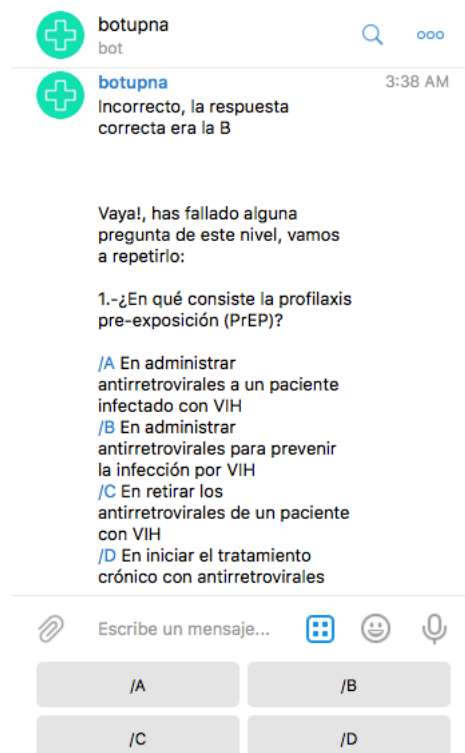


Figura 48 – Fallo de última pregunta de nivel

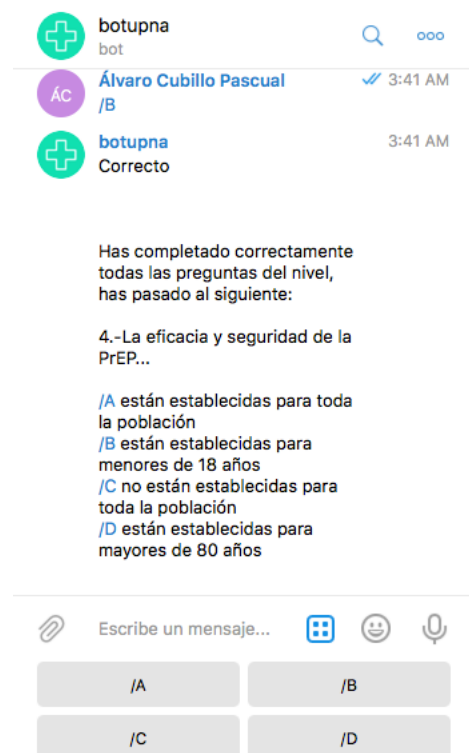


Figura 49 – Acierto de todas las preguntas de un nivel

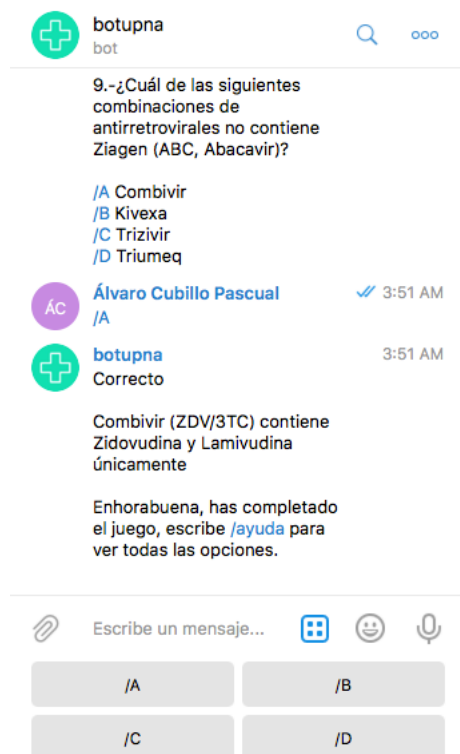


Figura 50 – Juego completado

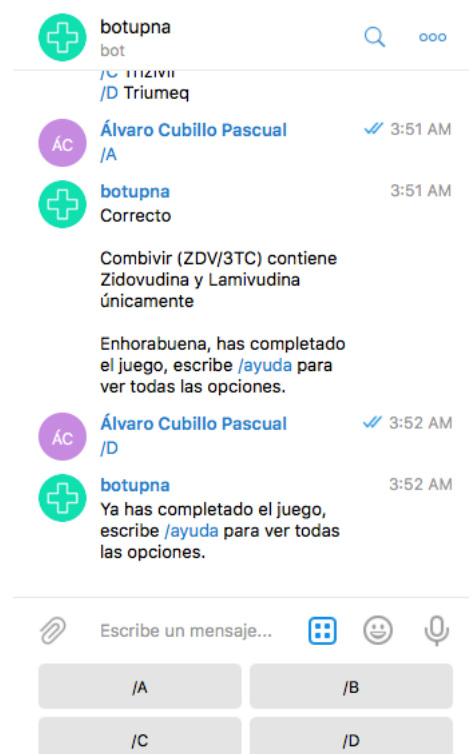


Figura 51 – Envío de "/A", "/B", "/C" o "/D" una vez finalizado el juego

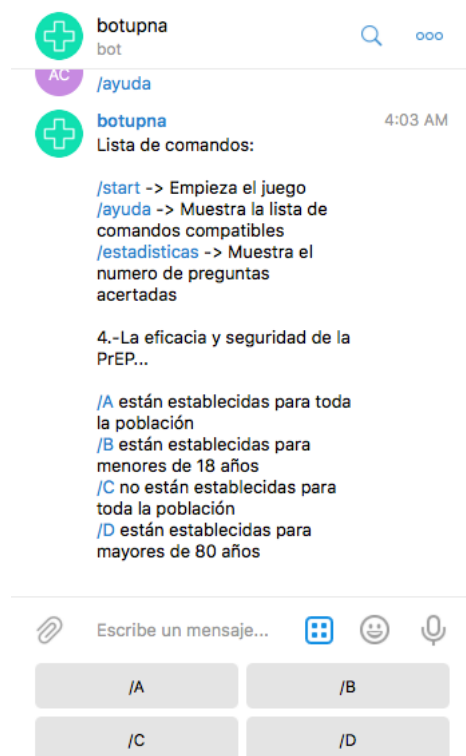


Figura 52 – Envío de "/ayuda" en cualquier momento del juego

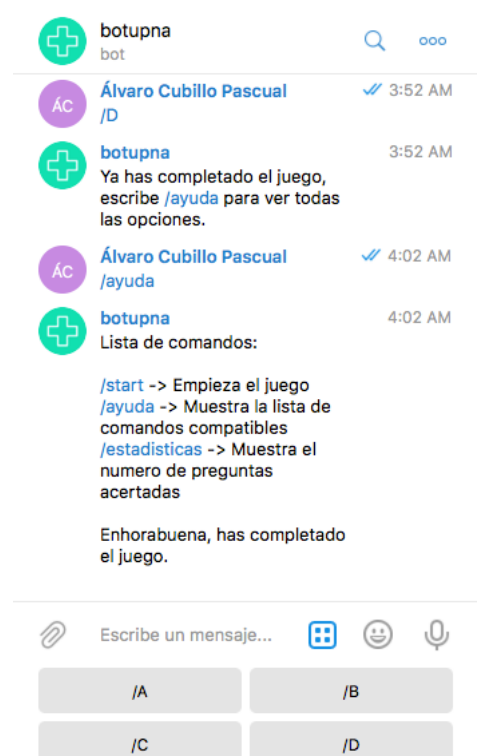


Figura 53 – Envío de "/ayuda/" una vez finalizado el juego

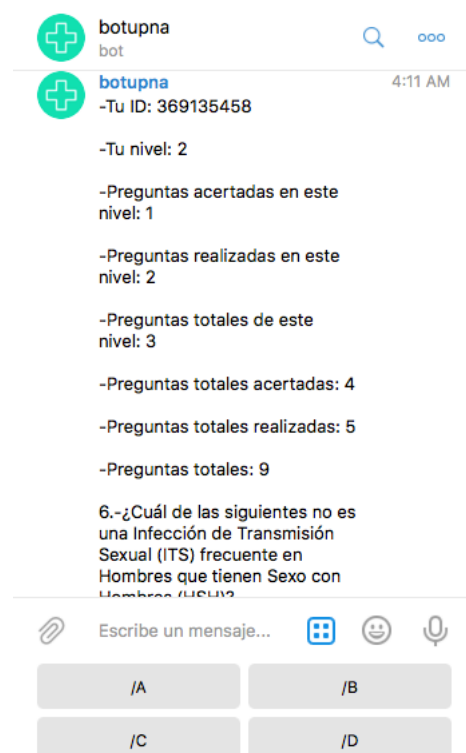


Figura 54 – Envío de "/estadísticas" en cualquier momento del juego

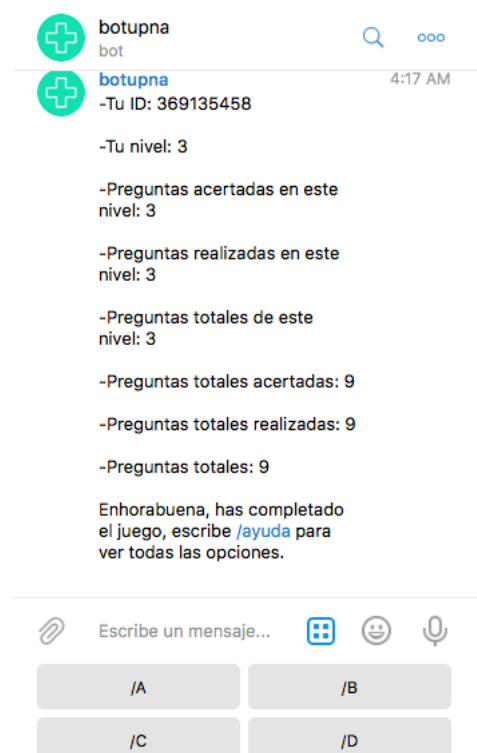


Figura 55 – Envío de "/estadísticas" una vez finalizado el juego

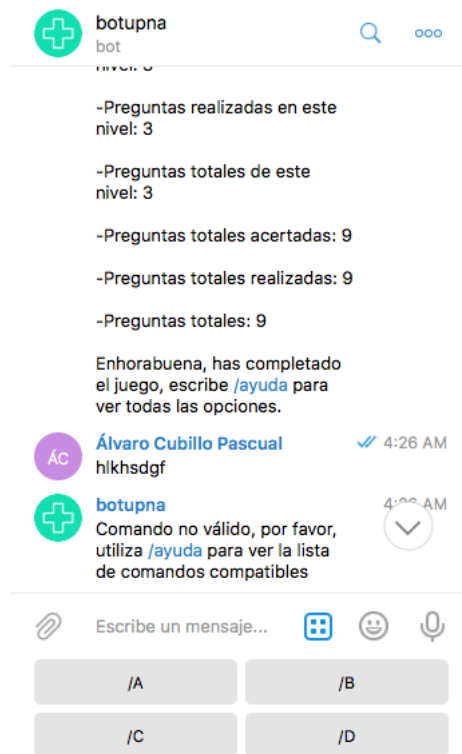


Figura 56 – Envío de un comando no válido



## 5. CONCLUSIONES

---

### 5.1. MÉDICAS

Hoy en día, con la tecnología tan presente en todas las facetas de nuestra vida deberíamos aprovecharla como una ventaja a la hora de fomentar la salud y prevenir enfermedades. Una aplicación como la que se ha desarrollado puede ser un pilar importante en las consultas dirigidas a pacientes con ETS, en especial con VIH, ya que se induce de manera interactiva al estudio y al conocimiento de la enfermedad que padecen y el tratamiento que reciben. Además, el hecho de que el facultativo pueda ver las estadísticas obtenidas por cada paciente le permite poder dirigir cada consulta de manera diferente, reforzando puntos débiles y evitando repetición de conceptos que ya se conocen. Todo ello en última instancia favorece la adherencia al tratamiento y la buena relación médico-paciente, puntos fundamentales a la hora de controlar en la mayor medida posible la epidemia de VIH que existe en nuestros días.

### 5.2. TECNOLÓGICAS

A nivel tecnológico se ha conseguido crear un bot funcional y totalmente automatizado, ya que el usuario prácticamente no tiene que escribir ningún comando. Esto se ha llevado a cabo con la implementación de un teclado, mostrando la lista de comandos al escribir “/” y reenviando la pregunta actual en el momento en que se deseen ver las estadísticas o la ayuda. De esta manera se pretende crear una interfaz fácil de usar para que el usuario utilice el bot con regularidad.

Como se ha diseñado la estructura del bot mediante niveles, impidiendo subir éstos sin haber completado el anterior, se tiene la seguridad de que al finalizar el juego el usuario ha contestado correctamente a todas las preguntas.

Mediante este sistema se pueden añadir, modificar o eliminar preguntas sin necesidad de modificar la estructura de las tablas de la base de datos.

Destacar también que los usuarios se añaden automáticamente una vez comiencen el juego, y que se puede conocer su “id” mediante el comando “/estadísticas”, para su posterior identificación.





## 6. TRABAJOS FUTUROS

---

A partir de la arquitectura diseñada se podría modificar la estructura del bot, eliminando los niveles, enviando todas las preguntas seguidas, y almacenando las respuestas de cada pregunta en otra tabla de la base de datos. Esto permitiría saber cuáles son los temas de la enfermedad que menos conoce cada usuario, pudiendo hacer más énfasis en ellos, pero tiene la limitación de que al añadir preguntas habría que modificar la estructura de la tabla de las respuestas, con lo que el personal médico tendría que aprender a utilizar algún programa de gestión de bases de datos.

Cabe destacar que es un programa fácilmente escalable ya que se pueden añadir, modificar o eliminar preguntas, se puede orientar a mas enfermedades o incluso cambiar totalmente la temática del bot.

Sería interesante desarrollar una evaluación en un entorno real, donde usuarios que padezcan ETS puedan probar el funcionamiento del bot y hacer un seguimiento de los conocimientos adquiridos gracias a éste. También preguntar qué les parece adquirir dichos conocimientos mediante la Gamificación, y si les resulta más entretenido que tener que aprender sólo de formas convencionales.



## 7. BIBLIOGRAFÍA

---

- [1] Ets.sobresexualidad.com. (2017). *Enfermedades de Transmisión Sexual | Otro sitio más de Sobre Sexualidad*. [online] Available at: <http://ets.sobresexualidad.com> [Accessed 19 Jun. 2017].
- [2] Cdc.gov. (2017). *Enfermedades de transmisión sexual*. [online] Available at: <https://www.cdc.gov/std/Spanish/default.htm> [Accessed 19 Jun. 2017].
- [3] Organización Mundial de la Salud. (2017). *VIH/SIDA*. [online] Available at: [http://www.who.int/topics/hiv\\_aids/es/](http://www.who.int/topics/hiv_aids/es/) [Accessed 19 Jun. 2017].
- [4] Cdc.gov. (2017). *CDC - ETS Hojas informativas*. [online] Available at: <https://www.cdc.gov/std/spanish/hojas-informativas.htm> [Accessed 19 Jun. 2017].
- [5] Organización Mundial de la Salud. (2017). *VIH/SIDA*. [online] Available at: <http://www.who.int/mediacentre/factsheets/fs360/es/> [Accessed 19 Jun. 2017].
- [6] Portnoy DB, Scott-Sheldon LAJ, Johnson BT, Carey MP. Computer-Delivered Interventions for Health Promotion and Behavioral Risk Reduction: A Meta-Analysis of 75 Randomized Controlled Trials, 1988 – 2007. *Prev Med*. 2008 Jul;47(1):3–16.
- [7] Bailey, J., Murray, E., Rait, G., Mercer, C., Morris, R., Peacock, R., Cassell, J. and Nazareth, I. (2017). *Interactive computer-based interventions for sexual health promotion*.
- [8] Pruijt H. Social interaction with computers: An interpretation of Weizenbaum's ELIZA and her heritage. *Soc Sci Comput Rev*. 2006 Nov 1;24(4):516–23.
- [9] Core.telegram.org. (2017). *Bots: An introduction for developers*. [online] Available at: <https://core.telegram.org/bots> [Accessed 17 Jun. 2017].
- [10] Ntaso.com. (2017). *Let's make a Telegram Bot with PHP | Chris on Code*. [online] Available at: <http://www.ntaso.com/lets-make-a-telegram-bot-with-php/> [Accessed 17 Jun. 2017].

- [11] R., S. (2017). *Keyboards - Telegram Bot API SDK*. [online] irazasyed.github.io. Available at: <https://irazasyed.github.io/telegram-bot-sdk/usage/keyboards/> [Accessed 17 Jun. 2017].
- [12] Devcenter.heroku.com. (2017). *Getting Started on Heroku with PHP / Heroku Dev Center*. [online] Available at: <https://devcenter.heroku.com/articles/getting-started-with-php#introduction> [Accessed 17 Jun. 2017].
- [13] López, D. (2017). *Bot en Telegram con PHP*. [online] Frostq - Blog de desarrollo web. Available at: [https://frostqui.github.io/bot\\_telegram\\_php](https://frostqui.github.io/bot_telegram_php) [Accessed 17 Jun. 2017].
- [14] Devcenter.heroku.com. (2017). *Getting Started on Heroku with PHP / Heroku Dev Center*. [online] Available at: <https://devcenter.heroku.com/articles/getting-started-with-php#provision-a-database> [Accessed 17 Jun. 2017].

